FINAL CODE

Apply ImageDataGenerator Functionality To Train And Test Set.IPYNB

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
# Training Datagen
train_datagen = ImageDataGenerator(rescale=1/255,zoom_range=0.2,horizontal_flip=True,vertical_flip=False)
# Testing Datagen
test_datagen = ImageDataGenerator(rescale=1/255)
```

```
# Training Dataset
x_train=train_datagen.flow_from_directory(r'/content/drive/MyDrive/Dataset/training_set',target_size=(64,64), class_mode='categorical',batch_size=900)
# Testing Dataset
x_test=test_datagen.flow_from_directory(r'/content/drive/MyDrive/Dataset/test_set',target_size=(64,64), class_mode='categorical',batch_size=900)
```

```
Found 15760 images belonging to 9 classes.
Found 2250 images belonging to 9 classes.
```

```
print("Len x-train : ", len(x_train))
print("Len x-test : ", len(x_test))
```

```
Len x-train :  18
Len x-test :  3
```

```
# The Class Indices in Training Dataset
x_train.class_indices
```

```
{'A': 0, 'B': 1, 'C': 2, 'D': 3, 'E': 4, 'F': 5, 'G': 6, 'H': 7, 'I': 8}
```

Import ImageDataGenerator Library And Configure It.IPYNB

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
# Training Datagen
train_datagen = ImageDataGenerator(rescale=1/255,zoom_range=0.2,horizontal_flip=True,vertical_flip=False)
# Testing Datagen
test_datagen = ImageDataGenerator(rescale=1/255)
```

File Edit Selection View Go Run Terminal Help — Add The Convolution Layer.ipynb - Visual Studio Code

Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

Add The Convolution Layer.ipynb | Add The Flatten Layer.ipynb | Add The Pooling Layer.ipynb | Adding The Dense Layers.ipynb | Fit And Save The Model.ipynb | Impo

D: > IBM > IBM-Project-1787-1658413612-main > IBM-Project-1787-1658413612-main > CODING PHASE > MODEL BUILDING > Add The Convolution Layer.ipynb > Real-Time Communication System Powered By AI For S

+ Code + Markdown ··· Select Kernel

[3] Python

```python
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

[4] Python

```python
# Training Datagen
train_datagen = ImageDataGenerator(rescale=1/255,zoom_range=0.2,horizontal_flip=True,vertical_flip=False)
# Testing Datagen
test_datagen = ImageDataGenerator(rescale=1/255)
```

[6] Python

```python
# Training Dataset
x_train=train_datagen.flow_from_directory(r'/content/drive/MyDrive/Dataset/training_set',target_size=(64,64), class_mode='categorical',batch_size=900)
# Testing Dataset
x_test=test_datagen.flow_from_directory(r'/content/drive/MyDrive/Dataset/test_set',target_size=(64,64), class_mode='categorical',batch_size=900)
```

```
Found 15760 images belonging to 9 classes.
Found 2250 images belonging to 9 classes.
```

```python
print("Len x-train : ", len(x_train))
print("Len x-test : ", len(x_test))
```

Restricted Mode ⊗ 0 ⚠ 0 Cell 1 of 11

Type here to search ENG 17:11 16-11-2022

File Edit Selection View Go Run Terminal Help — Add The Flatten Layer.ipynb - Visual Studio Code

Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

Add The Convolution Layer.ipynb | Add The Flatten Layer.ipynb | Add The Pooling Layer.ipynb | Adding The Dense Layers.ipynb | Fit And Save The Model.ipynb | Impo

M > IBM-Project-1787-1658413612-main > IBM-Project-1787-1658413612-main > CODING PHASE > MODEL BUILDING > Add The Flatten Layer.ipynb > Real-Time Communication System Powered By AI For Specially Abled

+ Code + Markdown ··· Select Kernel

```python
# Training Dataset
x_train=train_datagen.flow_from_directory(r'/content/drive/MyDrive/Dataset/training_set',target_size=(64,64), class_mode='categorical',batch_size=900)
# Testing Dataset
x_test=test_datagen.flow_from_directory(r'/content/drive/MyDrive/Dataset/test_set',target_size=(64,64), class_mode='categorical',batch_size=900)
```

[6] Python

```
Found 15760 images belonging to 9 classes.
Found 2250 images belonging to 9 classes.
```

```python
print("Len x-train : ", len(x_train))
print("Len x-test : ", len(x_test))
```

[7] Python

```
Len x-train :  18
Len x-test :  3
```

```python
# The Class Indices in Training Dataset
x_train.class_indices
```

[8] Python

```
{'A': 0, 'B': 1, 'C': 2, 'D': 3, 'E': 4, 'F': 5, 'G': 6, 'H': 7, 'I': 8}
```

Restricted Mode ⊗ 0 ⚠ 0 Cell 1 of 14

Type here to search ENG 17:11 16-11-2022

File Edit Selection View Go Run Terminal Help — Add The Pooling Layer.ipynb - Visual Studio Code

Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

Add The Convolution Layer.ipynb | Add The Flatten Layer.ipynb | Add The Pooling Layer.ipynb | Adding The Dense Layers.ipynb | Fit And Save The Model.ipynb | Impo

D: > IBM > IBM-Project-1787-1658413612-main > IBM-Project-1787-1658413612-main > CODING PHASE > MODEL BUILDING > Add The Pooling Layer.ipynb > Real-Time Communication System Powered By AI For Speci

+ Code + Markdown ··· Select Kernel

```
{'A': 0, 'B': 1, 'C': 2, 'D': 3, 'E': 4, 'F': 5, 'G': 6, 'H': 7, 'I': 8}
```

Model Creation

[9] Python

```python
# Importing Libraries
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense
```

+ Code + Markdown

[10] Python

```python
# Creating Model
model=Sequential()
```

[11] Python

```python
# Adding Layers
model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(64,64,3)))
```

[12] Python

```python
model.add(MaxPooling2D(pool_size=(2,2)))
```

Restricted Mode ⊗ 0 ⚠ 0 Cell 1 of 12

Type here to search ENG 17:12 16-11-2022

File Edit Selection View Go Run Terminal Help — Adding The Dense Layers.ipynb - Visual Studio Code

Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

Add The Convolution Layer.ipynb | Add The Flatten Layer.ipynb | Add The Pooling Layer.ipynb | Adding The Dense Layers.ipynb | Fit And Save The Model.ipynb | Impo

D: > IBM > IBM-Project-1787-1658413612-main > IBM-Project-1787-1658413612-main > CODING PHASE > MODEL BUILDING > Adding The Dense Layers.ipynb > Real-Time Communication System Powered By AI For Sp

+ Code + Markdown ··· Select Kernel

[11] Python

```python
# Adding Layers
model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(64,64,3)))
```

[12] Python

```python
model.add(MaxPooling2D(pool_size=(2,2)))
```

[13] Python

```python
model.add(Flatten())
```

[14] Python

```python
# Adding Dense Layers
model.add(Dense(300,activation='relu'))
model.add(Dense(150,activation='relu'))
model.add(Dense(9,activation='softmax'))
```

[15] Python

```python
# Compiling the Model
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
```

Restricted Mode ⊗ 0 ⚠ 0 Cell 1 of 15

Type here to search ENG 17:12 16-11-2022

File Edit Selection View Go Run Terminal Help — Fit And Save The Model.ipynb - Visual Studio Code

Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

Add The Convolution Layer.ipynb | Add The Flatten Layer.ipynb | Add The Pooling Layer.ipynb | Adding The Dense Layers.ipynb | Fit And Save The Model.ipynb

IBM-Project-1787-1658413612-main > IBM-Project-1787-1658413612-main > CODING PHASE > MODEL BUILDING > Fit And Save The Model.ipynb > Real-Time Communication System Powered By AI For Specially Abled

+ Code + Markdown ··· Select Kernel

```python
# Fitting the Model Generator
model.fit_generator(x_train,steps_per_epoch=len(x_train),epochs=10,validation_data=x_test,validation_steps=len(x_test))
```

[18] Python

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please
use `Model.fit`, which supports generators.


Epoch 1/10
18/18 [==============================] - 92s 5s/step - loss: 0.0049 - accuracy: 0.9994 - val_loss: 0.2635 - val_accuracy: 0.9773
Epoch 2/10
18/18 [==============================] - 90s 5s/step - loss: 0.0040 - accuracy: 0.9995 - val_loss: 0.2074 - val_accuracy: 0.9773
Epoch 3/10
18/18 [==============================] - 87s 5s/step - loss: 0.0041 - accuracy: 0.9995 - val_loss: 0.2460 - val_accuracy: 0.9773
Epoch 4/10
18/18 [==============================] - 91s 5s/step - loss: 0.0041 - accuracy: 0.9992 - val_loss: 0.2470 - val_accuracy: 0.9782
Epoch 5/10
18/18 [==============================] - 88s 5s/step - loss: 0.0037 - accuracy: 0.9993 - val_loss: 0.2439 - val_accuracy: 0.9782
Epoch 6/10
18/18 [==============================] - 88s 5s/step - loss: 0.0024 - accuracy: 0.9997 - val_loss: 0.2852 - val_accuracy: 0.9782
Epoch 7/10
18/18 [==============================] - 91s 5s/step - loss: 0.0023 - accuracy: 0.9997 - val_loss: 0.2589 - val_accuracy: 0.9782
Epoch 8/10
18/18 [==============================] - 93s 5s/step - loss: 0.0014 - accuracy: 1.0000 - val_loss: 0.2523 - val_accuracy: 0.9782
Epoch 9/10
```

Restricted Mode ⊗ 0 ⚠ 0 Cell 1 of 18

Type here to search 29°C ENG 17:12 16-11-2022

File Edit Selection View Go Run Terminal Help — Fit And Save The Model.ipynb - Visual Studio Code

Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

Add The Convolution Layer.ipynb | Add The Flatten Layer.ipynb | Add The Pooling Layer.ipynb | Adding The Dense Layers.ipynb | Fit And Save The Model.ipynb

IBM-Project-1787-1658413612-main > IBM-Project-1787-1658413612-main > CODING PHASE > MODEL BUILDING > Fit And Save The Model.ipynb > Real-Time Communication System Powered By AI For Specially Abled

+ Code + Markdown ··· Select Kernel

```
18/18 [==============================] - 91s 5s/step - loss: 0.0041 - accuracy: 0.9992 - val_loss: 0.2470 - val_accuracy: 0.9782
Epoch 5/10
18/18 [==============================] - 88s 5s/step - loss: 0.0037 - accuracy: 0.9993 - val_loss: 0.2439 - val_accuracy: 0.9782
Epoch 6/10
18/18 [==============================] - 88s 5s/step - loss: 0.0024 - accuracy: 0.9997 - val_loss: 0.2852 - val_accuracy: 0.9782
Epoch 7/10
18/18 [==============================] - 91s 5s/step - loss: 0.0023 - accuracy: 0.9997 - val_loss: 0.2589 - val_accuracy: 0.9782
Epoch 8/10
18/18 [==============================] - 93s 5s/step - loss: 0.0014 - accuracy: 1.0000 - val_loss: 0.2523 - val_accuracy: 0.9782
Epoch 9/10
18/18 [==============================] - 92s 5s/step - loss: 0.0013 - accuracy: 0.9999 - val_loss: 0.2269 - val_accuracy: 0.9778
Epoch 10/10
18/18 [==============================] - 91s 5s/step - loss: 0.0012 - accuracy: 0.9999 - val_loss: 0.2968 - val_accuracy: 0.9782

<keras.callbacks.History at 0x7fde26f54590>
```

Saving the Model

```python
model.save('asl_model_84_54.h5')
```

[19]

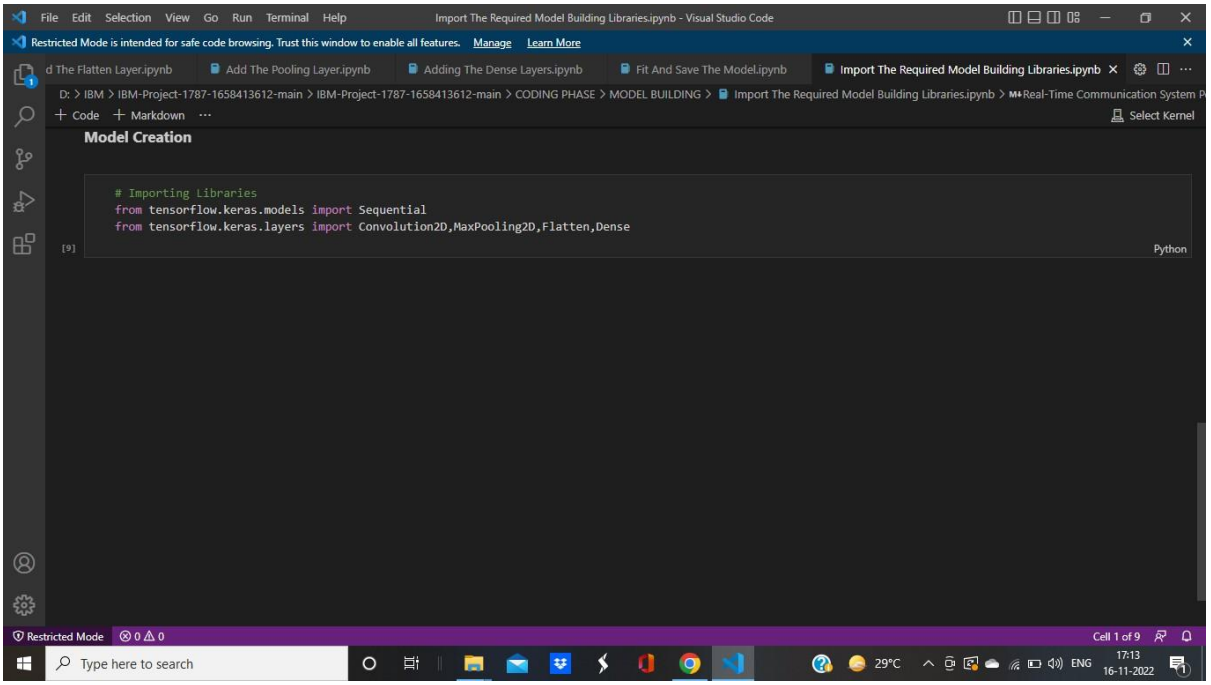
OneDrive

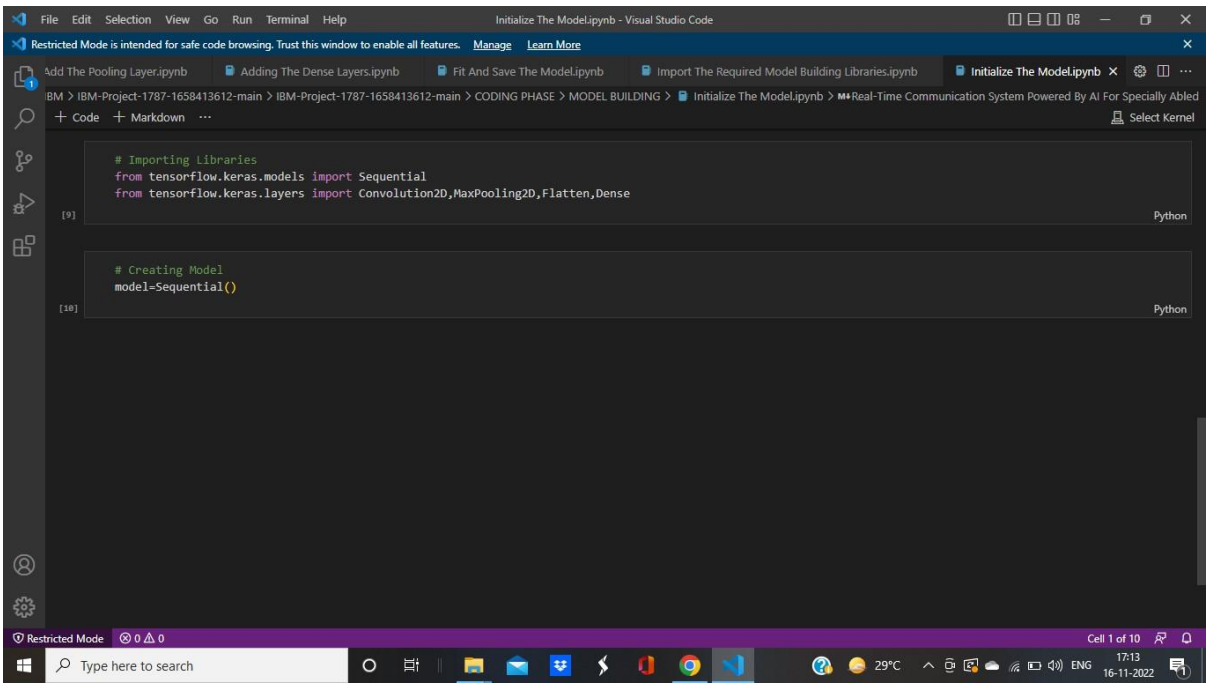
Screenshot saved

The screenshot was added to your OneDrive.

Restricted Mode ⊗ 0 ⚠ 0

Type here to search 29°C ENG 17:12 16-11-2022


File Edit Selection View Go Run Terminal Help
Import The Required Model Building Libraries.ipynb - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
d The Flatten Layer.ipynb
Add The Pooling Layer.ipynb
Adding The Dense Layers.ipynb
Fit And Save The Model.ipynb
Import The Required Model Building Libraries.ipynb
D: > IBM > IBM-Project-1787-1658413612-main > IBM-Project-1787-1658413612-main > CODING PHASE > MODEL BUILDING > Import The Required Model Building Libraries.ipynb > Real-Time Communication System P
+ Code + Markdown
Select Kernel
Model Creation
# Importing Libraries
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense
[9]
Python
Restricted Mode
Cell 1 of 9
Type here to search
29°C
ENG
17:13
16-11-2022


File Edit Selection View Go Run Terminal Help
Initialize The Model.ipynb - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More
Add The Pooling Layer.ipynb
Adding The Dense Layers.ipynb
Fit And Save The Model.ipynb
Import The Required Model Building Libraries.ipynb
Initialize The Model.ipynb
IBM > IBM-Project-1787-1658413612-main > IBM-Project-1787-1658413612-main > CODING PHASE > MODEL BUILDING > Initialize The Model.ipynb > Real-Time Communication System Powered By AI For Specially Abled
+ Code + Markdown
Select Kernel
# Importing Libraries
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense
[9]
Python
# Creating Model
model=Sequential()
[10]
Python
Restricted Mode
Cell 1 of 10
Type here to search
29°C
ENG
17:13
16-11-2022

Language used : Python

Required Packages :

Import numpy as np

Import cv2

Import os

From keras.models import load_model

From flask import flask, render_template, response

Import tensorflow as tf

From gtts import gtts #to convert text to speech

Global graph

Global writer

From skimage.transform import resize

App.py

```
from flask import Flask, Response, render_template
from camera import Video

app = Flask(__name__)
@app.route('/')
def index():
        return render_template('index.html')

def gen(camera):
        while True:
                frame = camera.get_frame()
                yield(b'--frame\r\n'
                        b'Content-Type: image/jpeg\r\n\r\n' + frame +
                        b'\r\n\r\n')

@app.route('/video_feed')
def video_feed():
        video = Video()
        return Response(gen(video), mimetype='multipart/x-mixed-replace; boundary =
frame')
```

```
if __name__ == '__main__':
        app.run()
```

main.py

```
import cv2

video = cv2.VideoCapture(0)

while True:
        ret, frame = video.read()
        cv2.imshow("Frame", frame)
        k = cv2.waitKey(1)
        if k == ord('q'):
                break

video.release()
cv2.destroyAllWindows()
```

Code to preprocess the frame captured from camera :

```
def detect(frame):
    img=resize(frame,(64,64,1))
    img=np.expand_dims(img,axis=0)
    if(np.max(img)>1):
        img=img/255.0
    with graph.as_default():
            prediction = model.predict_classes(img)
    print(prediction)
    pred=vals[prediction[0]]
```

camera.py

```
import cv2
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image

class Video(object):
    def __init__(self):
        self.video = cv2.VideoCapture(0)
        self.roi_start = (50, 150)
        self.roi_end = (250, 350)
        self.model = load_model('asl_model.h5') # Execute Local Trained Model
        # self.model = load_model('IBM_Communication_Model.h5') # Execute IBM
Trained Model
        self.index=['A','B','C','D','E','F','G','H','I']
        self.y = None
    def __del__(self):
        self.video.release()
    def get_frame(self):
        ret,frame = self.video.read()
        frame = cv2.resize(frame, (640, 480))
        copy = frame.copy()
```

```
        copy = copy[150:150+200,50:50+200]
        # Prediction Start
        cv2.imwrite('image.jpg',copy)
        copy_img = image.load_img('image.jpg', target_size=(64,64))
        x = image.img_to_array(copy_img)
        x = np.expand_dims(x, axis=0)
        pred = np.argmax(self.model.predict(x), axis=1)
        self.y = pred[0]
        cv2.putText(frame,'The Predicted Alphabet is:
'+str(self.index[self.y]),(100,50),cv2.FONT_HERSHEY_SIMPLEX,1,(0,0,0),3)
        ret,jpg = cv2.imencode('.jpg', frame)
        return jpg.tobytes()
```